IN THE CLAIMS:

1. (Currently Amended): A method for preventing premature shutdown of a virtual machine, comprising the steps of:

    starting a waiter thread in the virtual machine;

    registering daemon threads in a queue managed by the waiter thread;

    monitoring, by the waiter thread, the daemon threads ~~running in the virtual machine by the waiter thread~~ registered in the queue managed by the waiter thread; and

    preventing shutdown of the virtual machine so long as any of the monitored daemon threads are running in the virtual machine.

2. (Canceled)

3. (Currently Amended): The method as recited in claim [[2]] 1, further comprising the step of:

    responsive to a first daemon thread becoming inactive, searching for other inactive daemon threads registered in the queue.

4. (Currently Amended): The method as recited in claim [[2]] 1, further comprising the step of:

    responsive to a new daemon thread being created, appending the new daemon thread to the end of the queue.

5. (Currently Amended): The method as recited in claim [[2]] 1, further comprising the steps of:

    determining whether the queue is empty; and

    if the queue is empty, ~~waiting for the virtual machine to shut down or for a new daemon thread to be created~~ terminating the waiter thread.

6. (Currently Amended): The method as recited in claim 5, ~~further comprising the step of:~~

~~if a new daemon thread is created, adding the new daemon thread to the queue~~ wherein the virtual machine shuts down in response to termination of the waiter thread.

7. (Original): The method as recited in claim 1, wherein the waiter thread is a normal thread.

8. (Original): The method as recited in claim 1, wherein the daemon threads are created by remote method invocation code.

9. (Currently Amended): An apparatus for preventing premature shutdown of a virtual machine, comprising:

    starting means for starting a waiter thread in the virtual machine;

    registration means for registering daemon threads in a queue managed by the waiter thread;

    monitoring means for monitoring, by the waiter thread, the daemon threads ~~running in the virtual machine by the waiter thread~~ registered in the queue managed by the waiter thread; and

    preventing means for preventing shutdown of the virtual machine so long as any of the monitored daemon threads are running in the virtual machine.

10. (Canceled)

11. (Currently Amended): The apparatus as recited in claim [[10]] 9, further comprising:

    means, responsive to a first daemon thread becoming inactive, for searching for other inactive daemon threads registered in the queue.

12. (Currently Amended): The apparatus as recited in claim [[10]] 9, further comprising:

    means for appending the new daemon thread to the end of the queue, responsive to a new daemon thread being created.

13. (Currently Amended): The apparatus as recited in claim [[10]] 9, further comprising:

      determination means for determining whether the queue is empty; and

~~waiting means for waiting for the virtual machine to shut down or for a new daemon thread to be created,~~ means for terminating the waiter thread if the queue is empty.

14. (Currently Amended): The apparatus as recited in claim 13, ~~further comprising:~~

~~means for adding, if a new daemon thread is created, the new daemon thread to the queue~~ wherein the virtual machine shuts down in response to termination of the waiter thread.

15. (Original): The apparatus as recited in claim 9, wherein the waiter thread is a normal thread.

16. (Original): The apparatus as recited in claim 9, wherein the daemon threads are created by remote method invocation code.

17. (Currently Amended): A computer program product, in a computer readable medium, for preventing premature shutdown of a virtual machine, comprising:

      first instructions for starting a waiter thread in the virtual machine;

      second instructions for registering daemon threads in a queue managed by the waiter thread;

      ~~second~~ third instructions for monitoring, by the waiter thread, the daemon threads ~~running in the virtual machine by the waiter thread~~ registered in the queue managed by the waiter thread; and

      [[third]] fourth instructions for preventing shutdown of the virtual machine so long as any of the monitored daemon threads are running in the virtual machine.

18. (Canceled)

19. (Currently Amended): The computer program product as recited in claim [[18]] 17, further comprising:

      fifth instructions for responsive to a first daemon thread becoming inactive, searching for other inactive daemon threads registered in the queue.

20. (Currently Amended): The computer program product as recited in claim [[18]] 17, further comprising:

      sixth instructions for determining whether the queue is empty; and

      seventh instructions for ~~waiting for the virtual machine to shut down or for a new daemon thread to be created~~ terminating of the waiter thread if the queue is empty.